Електротехнички факултет

Универзитет у Београду

ПРОЈЕКАТ

На тему

Програмирање и контролисање Tello дрона у програмском језику Python

Предмет:

Теорија електричних кола

Ментор:

Проф. др Дејан Тошић,

редовни професор

Студенти:

Теодора Трмчић 21/0256

Илија Ристановић 21/0169

Тодор Александровић 21/0053

Милан Васић 21/0522

Београд, 2022.

Садржај

Списак слика:

1. Увод

1.1 Основни појмови

Дефиниција дрона: **Беспилотна летелица** (**БПЛ**; енгл. *unmanned aerial vehicle, UAV*) или популарно **дрон** (енгл. *drone - трут*), је летелица у којој нема пилота, посаде или путника, односно којим управља навигатор. Беспилотне летелице су компонента система беспилотне летелице (**СБПЛ**), који укључује додавање земаљског контролера и система комуникације са **БПЛ**. Лет беспилотних летелица може да функционише под даљинском контролом од стране људског оператера, као летелица са даљинским пилотирањем (РПА), или са различитим степенима аутономије, као што је помоћ аутопилота, до потпуно аутономних летелица које не захтевају људску интервенцију.

DJI Компанија: SZ DJI Technology Co., Ltd. или Shenzhen DJI Sciences and Technologies Ltd. (кинески: 深圳大疆创新科技有限公司) , познатији по трговачком имену DJI, што је скраћеница за Da-Jiang Innovations (енг. Great Frontier Innovations), је компанија са седиштем у Шенжену, граду познатијем као „кинеска Силицијумска Долина“, коју подржава неколико државних субјеката. DJI производи комерцијалне беспилотне летелице (дронове) за фотографисање и видео снимање из ваздуха. Такође дизајнира и производи карданске камере, акционе камере, стабилизаторе камера, платформе за лет, погонске системе и системе за контролу лета.

DJI Ryze Tello дрон је летелица малих димензија која представља улазни модел на тржишту. Веома је погодан за учење кроз широк опсег алата, укључујући и програмске библиотеке.

Карактеристике Tello дрона су:

Маса: 80 грама (са батеријом и пропелерима)

Димензије: 98×92.5×41 mm

Уграђене функције: Range Finder, Barometer, LED, Vision System,

2.4 GHz 802.11n Wi-Fi, 720p Live View

Максимална брзина лета: 8м/с

Максимална дистанца лета: 100м

Слика 1.1. Tello дрон.
(https://www.ryzerobotics.com/tello)

2. Формирање пројекта

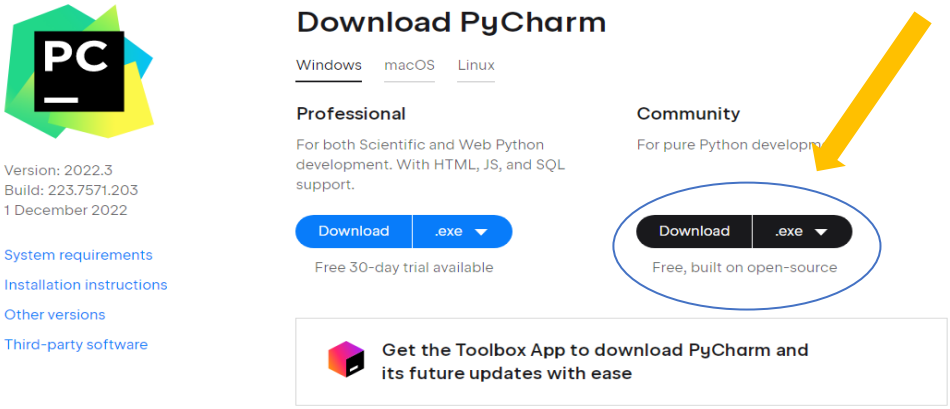
У оквиру предмета Теорија електричних кола, организована је израда пројекта едукативног типа. Сврха је реализовање функција које дрон извршава, на креативан начин користећи се софтверским алатима или електронским колима. Задатак у овом пројекту је да, користећи се искључиво софтверским алатом, дрон полети, правећи спиралу се подигне на висину и направи фотографију. У извештају је приказана реализација свих корака потребних за реализацију пројекта у програмском језику Python користећи окружење PyCharm.

2.1 Инсталација програмског окружења

Сам рад на пројекту започет је постављањем радног окружења. Програмски језик Python преузет је са сајта: https://www.python.org/downloads/. Препоручено је да се преузме најновија верзија која је доступна у тренутку израде.

Слика 2.1 Преузимање програмског језика Пајтон.

PyCharm се показао као један од најефикасније интегрисаних развојних окружења за програмирање у програмском језику Python. Инсталациони фолдер за програм налази се на линку: http://www.jetbrains.com/pycharm/download/. За рад на пројекту коришћена је Community верзија која је бесплатна свим корисницима.

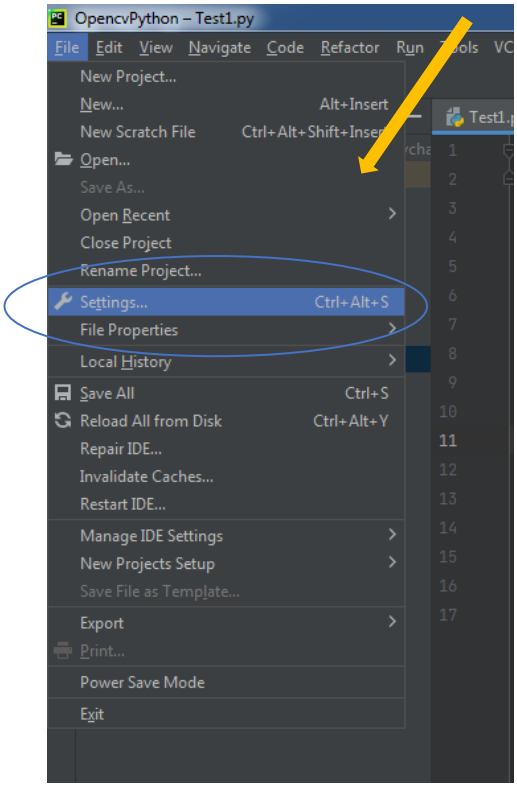


Слика 2.2 Преузимање развојног окружења Пај-Чарм.

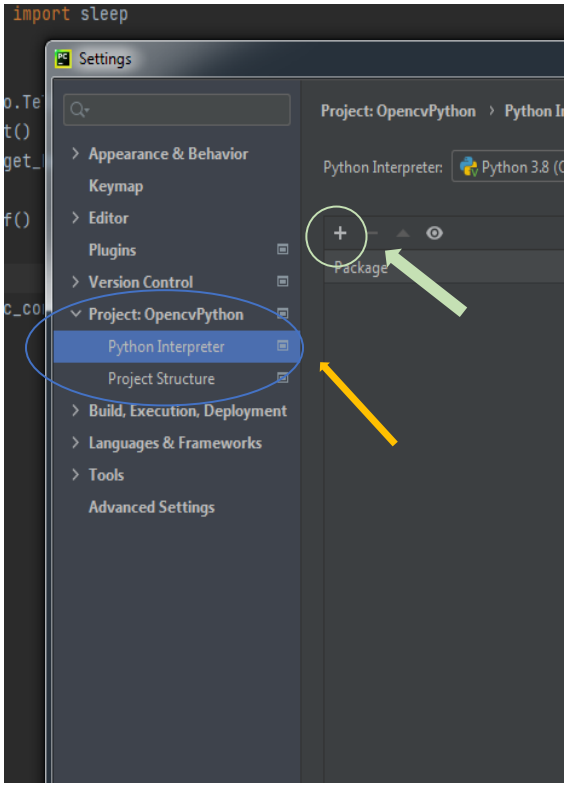
2.2 Подешавање програмског окружења

Након успешне инсталације програмског језика и програмског окружења, за комуникацију са дроном су потребне команде из библиотеке djitellopy. Додавање библиотеке врши се на следећи начин:

1. Потребно је креирати нови пројекат у којем ће се налазити кодови. Унутар креираног пројекта потребно је кликом на **FILE** отворити падајући мени на коме бирамо опцију **SETTINGS**. Након тога потребно је одабрати **Project: <име>** а затим отворити одељак **Python interpreter**.

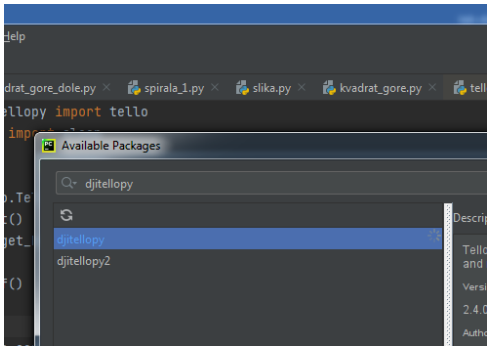


Слика 2.3 корак 1

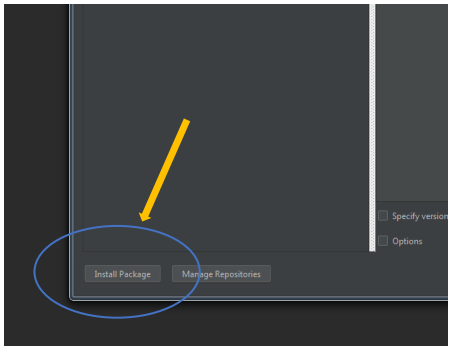


Слика 2.4 корак 1'

2. Потребно је кликом на **+** покренути додавање библиотеке. Унутар поља за претрагу потребно је написати **djitellopy**. На екрану се појављује тражена библиотека коју инсталирамо кликом на **Install Package**.



Слика 2.5 корак 2



Слика 2.6 корак 2'

Овим је наше окружење спремно за рад. Битно је нагласити да се подешавања односе само на пројекат који је тренутно отворен, те она неће утицати на остале пројекте. То такође значи да је приликом прављења неког следећег пројекта везаног за дрон потребно поново припремити библиотеку.

3. Програмски кôд

3.1 Делови кôда

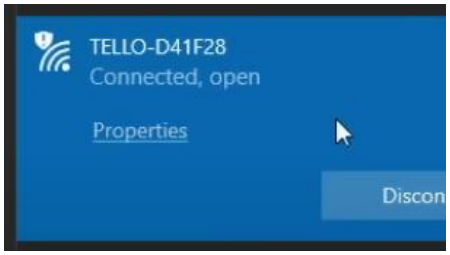
Све кодове потребно је писати у пројекту који смо припремили у претходном поглављу. Како бисмо могли да користимо готове функције за контролу дрона најпре позивамо:

```
from djitellopy import tello
```

Дрону приступамо помоћу функције **tello.Tello()**. Наш дрон називамо **dron**, и надаље то име користимо за позивање свих његових функција. Битно је обратити пажњу на коришћење великих и малих слова јер језик Python препознаје разлику. Додела је реализована као:

```
dron = tello.Tello()
```

Да бисмо успоставили конекцију између рачунара потребно је да најпре укључимо дрон на дугме за паљење и сачекамо пар секунди да он створи своју wifi приступну тачку. Након тога повезујемо наш рачунар на његову мрежу:



Унутар окружења позивамо команду **connect()**. Дата функција ће успоставити комуникацију са нашим дроном. Подешавање IP адресе је аутоматско.

Позив функције connect():

```
dron.connect()
```

Како бисмо започели лет користимо функцију takeoff(). Дрон се подиже у ваздух и стабилизује, након чега враћа потврду 'OK' или 'FALSE' у зависности од успешности полетања.

```
dron.takeoff()
```

За контролисање даљег управљања летом користимо функцију **send_rc_control()**. Дефиниција изгледа овако:

```
def send_rc_control(self, left_right_velocity: int, forward_backward_velocity: int, up_down_velocity: int,
                    yaw_velocity: int):
    """Send RC control via four channels. Command is sent every self.TIME_BTW_RC_CONTROL_COMMANDS seconds.
    Arguments:
        left_right_velocity: -100~100 (left/right)
        forward_backward_velocity: -100~100 (forward/backward)
        up_down_velocity: -100~100 (up/down)
        yaw_velocity: -100~100 (yaw)
    """
```

Параметри који се задају су брзине редом у правцима: лево-десно, напред-назад, горе-доле и окретање око своје вертикалне осе. Брзине се задају у опсегу [-100, 100] што представља брзину у центиметрима по секунди.

Програми функционишу тако да се након компајлирања и покретања кодови реализују веома брзо. Стога нам је потребан начин да правимо контролисану временску паузу између појединачних команди. То се решава коришћењем функције **sleep()** из библиотеке **time**, позивом:

```
from time import sleep
```

Команде коришћене у пројекту изгледају овако:

```
dron.send_rc_control(-15, 16, 35, 90)
sleep(12)
dron.send_rc_control(0,0,0,0)
```

Последња команда са свим нула параметрима служи како бисмо након извршавања прве функције (која у овом случају траје 12 секунди) зауставили дрон.

Слетање се реализује командом **land()**:

```
dron.land()
```

За прављење фотографије користи се функција **streamon()** која омогућује коришћење камере дрона. Прављење фотографије у одређеном тренутку реализујемо командом **get_frame_read().frame.** За коришћење датих функција потребно је увести библиотеке за слике и рад са фајловима преко **import cv2, shutil, os**. Коришћењем функција **imshow()** и **imwrite()** омогућујемо приказивање и чување фотографије. Приликом позива функције за чување, ствара се фолдер под задатим именом и у њега се смешта фотографија. Приликом наредног покретања програма постојећа фотографија се аутоматски брише и нова се смешта на њено место. Уз даљу оптимизацију коначни код којим се реализује прављење фотографије гласи:

```
import cv2

img = dron.get_frame_read().frame
img = cv2.resize(img, (1080, 720))
cv2.imshow("Image", img)
cv2.waitKey(1)
```


3.2 Коначан кôд

Кôд који је у целости коришћен за овај пројекат наведен је у наставку. У циљу демонстрације софистицираности дрона додата је команда **flip()** која резултује наглим маневром окретања дрона око своје осе за 360 степени. Код је могуће копирати и као готов користити за извршавање задатка.

```
from djitellopy import tello
from time import sleep, time
import cv2, shutil, os

file = os.getcwd() + "\pictures"
"""Folder pod imenom pictures ce se napraviti i u njemu ce se
pojaviti fotografija."""

if (os.path.exists(file)):
    shutil.rmtree(file)
os.mkdir(file)

dron = tello.Tello()
dron.connect()
print(dron.get_battery())

dron.streamon()

dron.takeoff()

dron.send_rc_control(-15, 16, 35, 90)
sleep(12)
dron.send_rc_control(0,0,0,0)
sleep(1)
dron.flip_forward()
sleep(1)

img = dron.get_frame_read().frame
img = cv2.resize(img, (1080, 720))
cv2.imshow("Image", img)
cv2.waitKey(1)
cv2.imwrite(file + f"/{time()}.jpg", img)

dron.send_rc_control(-15, 16, -35, 90)
sleep(7)
dron.send_rc_control(0,0,0,0)

dron.land()
```

Снимак лета дрона коришћењем овог кôда налази се на линку:
https://youtu.be/PX3XPx3BkqQ

3.3 Коначан кôд (верзија 2)

У циљу тестирања оптималног решења овог пројекта развијена је и верзија 2 програма који покреће дрон у линеаризованој путањи спирале. Кôд је приложен испод.

```
from djitellopy import tello
from time import sleep, time
import cv2, shutil, os

file = os.getcwd() + "\pictures"
"""Folder pod imenom pictures ce se napraviti i u njemu ce se pojaviti
fotografija."""

if (os.path.exists(file)):
    shutil.rmtree(file)
os.mkdir(file)

dron = tello.Tello()
dron.connect()
print(dron.get_battery())

gore = 35
okret = 45
napred = 30
levo=10

dron.streamon()
dron.takeoff()

dron.send_rc_control(levo, napred, gore, okret)
sleep(2)
dron.send_rc_control(0, 0, 0, 0)
sleep(1)

dron.send_rc_control(levo, napred, gore, okret)
sleep(2)
dron.send_rc_control(0, 0, 0, 0)
sleep(1)

dron.send_rc_control(levo, napred, gore, okret)
sleep(2)
dron.send_rc_control(0, 0, 0, 0)
sleep(1)

dron.send_rc_control(levo, napred, gore, okret)
sleep(2)
dron.send_rc_control(0, 0, 0, 0)
sleep(1)

dron.send_rc_control(levo, napred, gore, okret)
sleep(2)
dron.send_rc_control(0, 0, 0, 0)
sleep(1)

dron.send_rc_control(levo, napred, gore, okret)
sleep(2)
dron.send_rc_control(0, 0, 0, 0)
sleep(1)

img = dron.get_frame_read().frame
img = cv2.resize(img, (1080, 720))
cv2.imshow("Image", img)
cv2.waitKey(1)
cv2.imwrite(file + f"/{time()}.jpg", img)
dron.land()
```

4. Закључак

Израдом овог пројекта приказано је аутоматско контролисање дрона помоћу софтверског алата Python. Процес израде је корак по корак описан у сврху успешнијег понављања, као и разумевања функција које дрон реализује у реалном времену. Подизање дрона спиралним покретом на висину и прављење фотографије, као и управљање дроном аутоматски коришћењем софтвера, јесте једна широко применљива функција у електротехничкој струци, а такође и у комерцијалне и едукативне сврхе.